**DESIGN PRINCIPLES AND PATTERNS**

**Exercise 1: Implementing the Singleton Pattern**

**Logger.java**

package singleton;

public class Logger {

    private static Logger instance;

    private Logger() {

        System.out.println("Logger initialized.");

    }

    public static Logger getInstance() {

        if (instance == null) {

            instance = new Logger();

        }

        return instance;

    public void log(String message) {

        System.out.println("[LOG]: " + message);

    }

}

**Main.java**

package singleton;

public class Main {

    public static void main(String[] args) {

        Logger logger1 = Logger.getInstance();

        logger1.log("First log message");

        Logger logger2 = Logger.getInstance();

        logger2.log("Second log message");

        if (logger1 == logger2) {

            System.out.println("Only one Logger instance exists.");

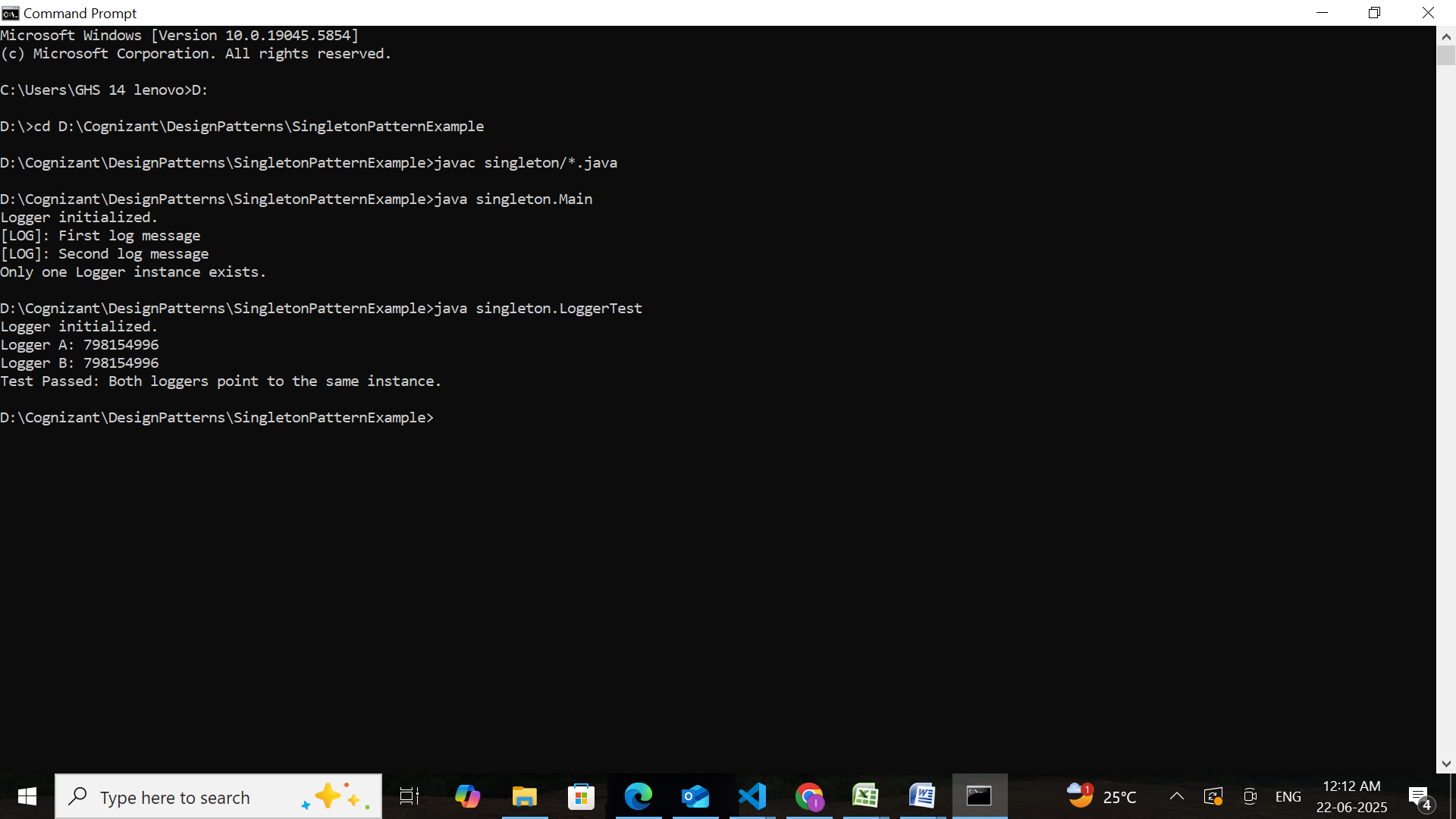
        } else {

            System.out.println("Different Logger instances found.");

        }

    }

}

**Output:**

**Exercise 2: Implementing the Factory Method Pattern**

**Document.java**

package document;

public interface Document {

    void open();

}

**ExcelDocument.java**

package document;

public class ExcelDocument implements Document {

    public void open() {

        System.out.println("Loading Excel spreadsheet workspace...");

    }

}

**PdfDocument.java**

package document;

public class PdfDocument implements Document {

    public void open() {

        System.out.println("Opening secured PDF file viewer...");

    }

}

**WordDocument.java**

package document;

public class WordDocument implements Document {

    public void open() {

        System.out.println("Launching Word file editor...");

    }

}

**DocumentFactory.java**

package factory;

import document.Document;

public abstract class DocumentFactory {

    public abstract Document createDocument();

}

**ExcelFactory.java**

package factory;

import document.Document;

import document.ExcelDocument;

public class ExcelFactory extends DocumentFactory {

    public Document createDocument() {

        return new ExcelDocument();

    }

}

**PdfFactory.java**

package factory;

import document.Document;

import document.PdfDocument;

public class PdfFactory extends DocumentFactory {

    public Document createDocument() {

        return new PdfDocument();

    }

}

**WordFactory.java**

package factory;

import document.Document;

import document.WordDocument;

public class WordFactory extends DocumentFactory {

    public Document createDocument() {

        return new WordDocument();

    }

}

**Main.java**

import document.Document;

import factory.DocumentFactory;

import factory.WordFactory;

import factory.PdfFactory;

import factory.ExcelFactory;

public class Main {

    public static void main(String[] args) {

        DocumentFactory wordFactory = new WordFactory();

        Document wordDoc = wordFactory.createDocument();

        wordDoc.open();

        DocumentFactory pdfFactory = new PdfFactory();

        Document pdfDoc = pdfFactory.createDocument();

        pdfDoc.open();

        DocumentFactory excelFactory = new ExcelFactory();

        Document excelDoc = excelFactory.createDocument();

        excelDoc.open();

    }

}

**Output:**

